

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pada akhir 2013, ada 9,1 miliar unit perangkat *Internet of Things* (IoT) dengan konektivitas *Internet Protocol* dan berkomunikasi tanpa interaksi dengan manusia, *Internasional Data Corporation (IDC)* memperkirakan pertumbuhan IoT yang diterapkann tiap tahun mencapai 17.5% diperkirakan menjadi 28,1 miliar di tahun 2020 [1]. Bahkan *Cisco* mempunyai predeksi dua kali lipat lebih besar yaitu 50 miliar pada tahun 2020 [2]. Contoh perangkat IoT yang saling terhubung dan berkomunikasi melalui internet adalah sistem pemantauan dan kontrol *Heating, Ventilation, and Air Conditioning* pada rumah pintar, transportasi, kesehatan, pertanian, otomasi industri, dan tanggap darurat bencana[3].

Perangkat IoT yang digunakan diberbagai tempat sering dianggap sebagai sistem yang tidak perlu mengubah *requirements* dan fungsinya, namun, pada kenyataanya dimana perangkat IoT ini berjalan pasti akan berubah[4]. Perubahan ini meliputi perubahan *behavior*, parameter yang terkait komunikasi dengan sistem lain atau pengguna, memperbaiki kesalahan, bisa masalah keamanan, yang dilaporkan pengguna setelah perangkat IoT digunakan[4].

Berbagai perubahan perangkat IoT dapat dilakukan dengan mengganti *firmware*, untuk mengganti *firmware* pada perangkat IoT harus keluar mengambil perangkat IoT, menghubungkan ke komputer, kemudian melakukan *update* dan mengembalikan perangkat IoT ke tempat. Namun, hal ini tidak dapat terus dilakukan bagi perusahaan yang memiliki perangkat IoT di berbagai tempat, seperti yang dilakukan *Chrysler* “merek mobil” pada tahun 2015 mereka dikritik karena mengirim perangkat *flashdisk* kepelangan untuk melakukan *update firmware* karena sangat rentan, *flashdisk* dapat diambil, di modifikasi dan dikirim kembali[5].

Jika sistem pada perangkat IoT sudah dapat berkomunikasi melalui antarmuka jaringan, hal ini bisa dimanfaatkan untuk menerapkan pembaruan *firmware* pada sistem IoT yang disebut dengan *Over the air (OTA)*[4]. OTA di lakukan oleh *Tesla* pada tahun tahun 2016 mengirimkan pembaruan *firmware* pada mobil mereka dan konsumen dapat mengatur akan melakukan pembaruan pada saat mobil di parkir [5].

Over the air update adalah proses memuat *firmware* pada modul ESP perangkat IoT menggunakan koneksi jaringan Wi-Fi dari pada menggunakan kabel *port serial* [6]. Secara umum isitilah *over the air* adalah mekanisme penggunaan *wireless* untuk mengirim data, memperbarui paket untuk pembaruan *firmware* atau perangkat lunak ke perangkat *mobile*, sehingga pengguna tidak perlu pergi mengakses fisik perangkat untuk mengubah aplikasi, parameter, *firmware*, atau memperbarui *software* [7].

Over the air update pada perangkat IoT sudah ada di pasaran dalam produk merek *Libelium*, namun *OTA* hanya bisa dilakukan pada perangkat IoT *Libelium* melalui *File Transfer Protocol* (FTP). Selain itu ada *particle.io* sama seperti *Libelium* hanya bisa digunakan untuk perangkat IoT yang mereka sediakan tidak bisa untuk perangkat lain. Selain dua perangkat berbayar *libelium* dan *particle.io*, *OTA* di sediakan oleh *espresif* melalui produk bernama *esp8266 12-E*, dapat melakukan update *firmware* melalui *Arduino ide*, web browser, dan *HTTP Server*[6].

Over the air update dengan *esp8266* yang telah tersedia melalui *arduino ide* dan *web browser* memiliki keterbatasan hanya bisa dilakukan dalam satu jaringan yang sama dan *HTTP Server* bisa dilakukan dengan jaringan berbeda [6]. Namun pada penelitian pengujian sensing data suhu dan kelembapan pada penelitian sebelumnya menunjukkan protokol HTTP 6 kali lebih lambat melakukan *transfer* data dari pada protokol MQTT dalam 60 detik dalam 5 kali percobaan di dapatkan rata-rata HTTP 934.4 data dan MQTT 6520.2 data [8]. Dalam IoT komunikasi dilakukan dengan berbagai protokol, misal CoAP, AMQP, MQTT, XMPP, dan HTTP [9].

Protokol MQTT adalah protokol pesan *publish/subscribe*, sangat sederhana ringan dan dirancang untuk perangkat IoT yang sangat terbatas oleh jaringan dengan *bandwidth* kecil, memiliki latensi tinggi. Prinsip desain protokol MQTT adalah untuk meminimalkan penggunaan *bandwidth* pada jaringan dan kebutuhan sumber daya atau penggunaan baterai pada perangkat IoT, sambil berusaha memastikan kehandalan untuk memastikan pengiriman data benar-benar terkirim *Quality of Service (QoS)* [10]. Contoh penggunaan protokol MQTT adalah Facebook Messenger[11] pada awal peluncurannya tahun 2011.

Berdasarkan latar belakang tersebut akan dilakukan uji coba implementasi protokol MQTT yang dapat melakukan *over the air update* pada perangkat *IoT*. *over the air update* dengan *protokol MQTT* akan digunakan untuk melakukan *update firmware, software* perangkat *IoT*, dari jaringan lokal atau jaringan internet, melalui media aplikasi berbasis website. Aplikasi berbasis website sebagai media *interface* pengguna untuk melakukan *update file firmware* dan melakukan monitoring hasil *upload* apakah berhasil atau tidak. Protokol MQTT digunakan sebagai media pengiriman *file firmware* hasil *build* dari *Arduino IDE* berupa *file tipe bin*, yang di *publish* ke perangkat *IoT* yang telah melakukan *subscribe* pada suatu *topic*.

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas maka didapat rumusan masalah sebagai berikut:

- a. Bagaimana mengimplemntasikan protokol MQTT untuk melakukan *over the air update* ?
- b. Bagaimana membuat aplikasi berbasis website sebagai media interaksi pengguna untuk melakukan *upload file firmwate tipe bin* dari *build arduino ide*, lalu melakukan *PUBLISH* ke perangkat *IoT* yang menjadi *SUBSCRIBER*?
- c. Bagaimana melakukan evaluasi efektivitas penggunaan protokol MQTT untuk melakukakan *over the air update firmware* atau pemrograman ulang perangkat *IoT* melalui jaringan lokal dan internet ?

1.3. Tujuan Penelitian

Tujuan penelitian tugas akhir rancang bangun layanan *over the air update firmware* menggunakan protokol MQTT pada *IoT* adalah untuk mengimplementasikan *over the air update firmware* pada perangkat *IoT* dengan *protocol MQTT*, membuat aplikasi berbasis website yang dapat publish *file tipe bin* ke perangkat *IoT* yang telah menjadi *subscriber* dan melakukan evaluasi efektivitas penerapan *OTA* dengan *protokol MQTT*.

1.4. Batasan Masalah

Batasan Masalah dilakukan agar dapat fokus ke inti tujuan yang ingin dicapai, dengan batasan masalah penelitian sebagai berikut.

- a. Menggunakan sensor DHT11 untuk uji coba bahwa program yang berjalan pada perangkat IoT berjalan sesuai.
- b. Hanya menggunakan mikrokontroler *NodeMCU esp8266-12E*, sebagai *mikrokontroler* pada perangkat IoT.
- c. Uji coba menggunakan 2 perangkat IoT, yang telah terhubung dengan sensor DHT11.
- d. Membangun layanan untuk *over the air update*, dengan aplikasi berbasis website sebagai media interaksi pengguna.
- e. Media penyimpanan hanya untuk menyimpan data login pengguna dan data *topic*.
- f. Tidak menyimpan data hasil *sensing*.
- g. Hanya menggunakan standar keamanan yang tersedia pada *protocol* MQTT.

